Optimal Path Following for Differentially Flat Robotic Systems through a Geometric Problem Formulation

Wannes Van Loock, Goele Pipeleers, Moritz Diehl, Joris De Schutter and Jan Swevers

***Abstract*—Path following deals with the problem of following a geometric path without any predefined timing information and constitutes an important step in solving the motion planning problem. For differentially flat systems, it has been shown that the projection of the dynamics along the geometric path onto a linear single-input system leads to a small dimensional optimal control problem. Although the projection simplifies the problem to great extent, the resulting problem remains difficult to solve, in particular in the case of nonlinear system dynamics and time-optimal problems. This paper proposes a nonlinear change of variables, using a time transformation, to arrive at a fixed end-time optimal control problem. Numerical simulations on a robotic manipulator and a quadrotor reveal that the proposed problem formulation is solved efficiently without requiring an accurate initial guess.**

***Index Terms*—Optimal Control, Differential Flatness, Nonholonomic Motion Planning.**

I. Introduction

As opposed to trajectory tracking where a reference specified in time is tracked, path following deals with the problem of following a geometric path without any preassigned timing information. Many industrial tasks are in fact path following problems, such as robot path following in e.g. welding or painting [1]–[4], control of autonomous vehicles [5], [6], but also control of chemical reactors [7] and batch crystallization [8]. In addition path following is often considered as the low level stage in a decoupled motion planning problem [1], [3]. In a first step, a high level planner determines a geometric path accounting only for geometric path constraints. Subsequently, an (optimal) velocity profile along the geometric path that takes into account the system dynamics and limitations, such as actuator saturation, is determined.

Most path following methods employ the fact that motion along a path can be described by a single coordinate, commonly denoted by s. For differentially flat systems the dynamics are projected along the path onto a single-input system, by applying the chainrule. Using this projection, [9] derive a small dimensional optimal control problem for an overhead crane. In [7] this approach is generalized for differentially flat systems. Furthermore, the authors provide easy-to-check conditions for determining whether a path is exactly followable. Despite its small dimension, the optimization problem remains difficult to solve and requires a good initialization, especially for free end-time problems. Moreover, the optimization variables enter through the definition of the path into the optimization problem. Hence, even for linear systems, a nonlinear reference path quickly complicates the problem, as shown in Section III.

This research was supported by Research Council KUL: BOF PFV/10/002 Optimization in Engineering Center OPTEC, GOA/10/11 Global real-time optimal control of autonomous robots and mechatronic systems and the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office (DYSCO). FWO projects: G.0320.08 (convex MPC), G.0377.09 (Mechatronics MPC); EU: FP7- EMBOCON (ICT-248940), FP7-SADCO (MC ITN-264735), ERC ST HIGHWIND (259 166). Goele Pipeleers is a Postdoctoral Fellow of the Research Foundation – Flanders (FWO – Vlaanderen).

W. Van Loock, G. Pipeleers, J. De Schutter and J. Swevers (email: wannes.vanloock@kuleuven.be; goele.pipeleers@kuleuven.be; joris.deschutter@kuleuven.be; jan.swevers@kuleuven.be) are with the department of mechanical engineering, division PMA, KU Leuven, Celestijnenlaan 300B, 3001 Leuven

M. Diehl (email: moritz.diehl@kuleuven.be) is with the institute of microsystem engineering (IMTEK), University of Freiburg, Georges-Köhler-Allee 102. D-79110 Freuburg, and the department of electrical engineering (ESAT), KU Leuven, Kasteelpark Arenberg 10, 3001 Leuven.

These problems are largely overcome by assuming a nonnegative velocity along the path such that time can be eliminated from consideration and the coordinate s can be used as independent variable [10], thus, arriving at a geometric problem formulation. A similar transformation is used in [2]. In their work a clever time transformation renders the time-optimal path following problem convex for the case of a simplified robotic manipulators. In [4], the authors apply sequential convex programming to broaden the set of models and constraints. The method however relies on finding a convex-concave decomposition of the constraints, which is difficult in many cases, hereby limiting its applicability. Our contribution presents a generalization of the approach of [2] for differentially flat systems using the results from [7]. Although in general the convexity of the problem is lost, the proposed problem formulation remains appealing since (i) the problem dimension remains small, (ii) the optimization variables no longer enter through the definition of the path, (iii) the problem is transformed into a fixed end-time problem and (iv) numerical experiments reveal that the solver no longer requires an accurate initial guess and computation times remain comparable to the convex problem as obtained in [2]. Moreover, the proposed problem formulation allows for an intuitive understanding of the conditions for which a path is exactly followable. Furthermore, a software package is released that simplifies the modeling of path following problems and can be downloaded from http://www.kuleuven.be/optec/software/.

Section II of this paper describes the notion of differentially flat systems. In Section III the considered problem is first presented for general nonlinear systems. Afterwards, the problem formulation from [7], which is based on projecting the dynamics along the path, is briefly reviewed. Section IV details our main contribution. A time transformation is proposed that transforms the problem into a fixed end-time optimal control problem for which the variables no longer enter through the definition of the path. It is shown how to cope with singularities that arise in this new formulation. Furthermore, conditions for which a path is exactly followable are reviewed and an intuitive proof is presented. Numerical experiments in Section V illustrate the efficiency of our approach on a robotic manipulator and a quadrotor. Finally, in Section VI we discuss other applications, not only in the field of robotics and draw conclusions.

In the following, we will use $\partial_\tau^k g(\tau)$ to denote the k-th derivative of $g(\tau)$ with respect to τ. For $k = 1$ we will simply write $\partial_\tau g(\tau)$. We will use the shorthand notation $\dot{g}, \ddot{g}$ to denote the first and second derivative with respect to time. I_n denotes the $n \times n$ identity matrix.

II. Differential flatness

Before addressing the problem statement, let us first recall the notion of differential flatness, which was initially introduced in [11].

A system is said to be differentially flat if there exists a set of variables, commonly referred to as flat output, such that every other system variable is a function of the flat output and a finite number of its time derivatives. More precisely, the system

$$\dot{x} = f(x, u) \tag{1}$$

with $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ is differentially flat if there exists a set of variables $y \in \mathbb{R}^m$, such that

$$y = g(x, u, \dots, \partial_t^l u)$$

and

$$\begin{aligned} x &= \Phi(y, \dots, \partial_t^{k-1} y) \\ u &= \Psi(y, \dots, \partial_t^k y), \end{aligned} \tag{2}$$

for some $l, k \in \mathbb{N}$.

The flat output characterizes all the state space motions and corresponding input history. It is a powerful concept in motion planning as it avoids having to integrate the nonlinear differential equations. The interested reader is invited to refer to [12]–[18] for further readings concerning flatness and motion planning.

III. Problem statement

Consider a geometric reference $y_r(s) \in \mathcal{C}^k$, a parametrized curve as a function of a scalar path coordinate s, for the flat output. The time dependency follows from $s(t)$. Without loss of generality, it is assumed that the trajectory starts at $t = 0$, ends at $t = T$ and $s(0) = 0 \leq s(t) \leq s(T) = 1$. Furthermore, it is assumed that the velocity along the path is non-negative, i.e. $\dot{s}(t) \geq 0$, that the system performs a rest-to-rest motion and that the boundary conditions x_0, x_T are consistent with the reference path, such that $x_0 = \Phi(y_r(s(0)), 0, \dots, 0)$ and $x_T = \Phi(y_r(s(T)), 0, \dots, 0)$.

In this paper, the goal is to determine an input signal $u(t)$ such that (i) the geometric reference is followed exactly by the flat output, i.e.

$$y(t) = y_r(s(t)), \tag{3a}$$

(ii) constraints on states and inputs

$$x(t) \in \mathcal{X} \text{ and } u(t) \in \mathcal{U}, \tag{3b}$$

are respected and (iii) the cost function

$$J = T + \int_0^T F(x(\tau), u(\tau)) d\tau \tag{3c}$$

is minimized, where the duration T is weighed against a performance criterion $F(\cdot)$, e.g. energy consumption or other regularization terms.

For an arbitrary nonlinear system this problem is challenging to solve as it requires integrating a nonlinear state space model and imposing the algebraic equations $y(t) = y_r(s(t))$. However, when considering differentially flat systems, the system dynamics can be projected along the path onto a linear single-input system that is trivial to integrate [7]. By applying the chainrule, we find the time derivatives of $y_r(s(t))$

$$\begin{aligned} \dot{y}_r &= \partial_s y_r \dot{s}, \\ \ddot{y}_r &= \partial_s^2 y_r \dot{s}^2 + \partial_s y_r \ddot{s}, \end{aligned} \tag{4}$$

and so on. Then by substituting (4) into (2) we rewrite the states and inputs of the system as

$$\begin{aligned} x &= \Phi_s(s, \dot{s}, \dots, \partial_t^{k-1} s) = \Phi_s(\sigma) \\ u &= \Psi_s(s, \dot{s}, \dots, \partial_t^{k} s) = \Psi_s(\sigma, v), \end{aligned}$$

where $\sigma(t) = (\sigma_0, \dots, \sigma_{k-1})^T = (s, \dot{s}, \dots, \partial_t^{k-1} s)^T$ and $v(t) = \partial_t^k s$. The rest-to-rest path following problem can now be reformulated as the smaller dimensional optimal control problem with states σ and control v

$$\begin{aligned} \underset{\sigma(\cdot), v(\cdot), T}{\text{minimize}} \quad & T + \int_0^T F_s(\sigma(\tau), v(\tau)) d\tau \\ \text{subject to} \quad & \dot{\sigma}(t) = \begin{pmatrix} 0 & I_{k-1} \\ 0 & 0 \end{pmatrix} \sigma(t) + (0, \dots, 0, 1)^T v(t) \\ & \sigma(0) = (0, \dots, 0)^T,\ \sigma(T) = (1, 0 \dots, 0)^T \\ & T \geq 0 \\ & \sigma_1(t) \geq 0, \forall t \in [0, T] \\ & \Phi_s(\sigma(t)) \in \mathcal{X},\ \forall t \in [0, T] \\ & \Psi_s(\sigma(t), v(t)) \in \mathcal{U},\ \forall t \in [0, T], \end{aligned} \tag{5}$$

Although the projection simplifies the problem (3) to a great extent, it suffers from some important drawbacks.

1) The variable $\sigma_0 = s$ enters the optimization problem through $y_r(s(t))$. Problems with nonlinear paths will introduce more nonlinearity into the problem.
2) The problem is a free end-time optimal control problem for which the solution can vary quite nonlinearly with changes in T, which usually results in slower convergence compared to fixed end-time problems.
3) Due to the above two reasons, an accurate initial guess is often needed to ensure convergence.

The following section, in which we show how to overcome these difficulties by a transformation of variables, details our main contribution.

IV. Time transformation

The key idea is to transform the problem such that, instead of the time t, the path coordinate s becomes the independent variable. In this way, the problem is transformed into a fixed end-time optimal control problem. Moreover, the optimization variables no longer enter the problem through the reference path $y_r(s)$. The transformation, proposed in [2], consists of parameterizing the velocity along the path as a function of the path coordinate:

$$b(s) := \dot{s}^2. \tag{6}$$

By taking the time derivative on both sides of (6), we obtain $\partial_s b(s) \dot{s} = 2\ddot{s}\dot{s}$ or

$$\ddot{s} = \frac{\partial_s b(s)}{2}.$$

We also require derivatives of higher order, which are obtained by repeatedly applying the chainrule

$$\begin{aligned} \partial_t^3 s &= \frac{\partial_s^2 b(s) \dot{s}}{2} = \frac{\partial_s^2 b(s) \sqrt{b(s)}}{2}, \\ \partial_t^4 s &= \frac{\partial_s^2 b(s) \ddot{s}}{2} + \frac{\partial_s^3 b(s) \dot{s}^2}{2} = \frac{\partial_s^2 b(s) \partial_s b(s)}{4} + \frac{\partial_s^3 b(s) b(s)}{2}, \end{aligned}$$

and so on.

States and inputs can now be reformulated as a function of s and $b(s)$ and its derivatives:

$$\begin{aligned} x &= \Phi_b(s, b, \partial_s b, \dots, \partial_s^{k-2} b) = \Phi_b(s, \beta) \\ u &= \Psi_b(s, b, \partial_s b, \dots, \partial_s^{k-1} b) = \Psi_b(s, \beta, w), \end{aligned}$$

where $\beta(s) = (\beta_0, \dots, \beta_{k-2})^T = (b, \partial_s b, \dots, \partial_s^{k-2} b)^T$ and $w(s) = \partial_s^{k-1} b$. Furthermore, using $dt = \frac{ds}{\dot{s}} = \frac{ds}{\sqrt{b}}$, the objective function is reformulated as

$$J = \int_0^T 1 + F(x(\tau), u(\tau)) d\tau = \int_0^1 \frac{1 + F_b(s, \beta(s), w(s))}{\sqrt{\beta_0(s)}} ds. \tag{7}$$

With the transformation (6), problem (5) is reformulated as the optimal control problem with pseudotime s, differential states β and control w

$$\begin{aligned} \underset{\beta(\cdot), w(\cdot)}{\text{minimize}} \quad & \int_0^1 \frac{1 + F_b(s, \beta(s), w(s))}{\sqrt{\beta_0(s)}} ds \\ \text{subject to} \quad & \partial_s \beta(s) = \begin{pmatrix} 0 & I_{k-2} \\ 0 & 0 \end{pmatrix} \beta(s) + (0, \dots, 0, 1)^T w(s) \\ & \beta_0(s) \geq 0, \forall s \in [0, 1] \\ & \Phi_b(s, \beta(s)) \in \mathcal{X},\ \forall s \in [0, 1] \\ & \Psi_b(s, \beta(s), w(s)) \in \mathcal{U},\ \forall s \in [0, 1]. \end{aligned} \tag{8}$$

The initial and terminal values for β are omitted deliberately. Instead, a rest-to-rest motion is imposed by requiring $\partial_s^i y_r(0) = 0$ for $i = 1, \dots, k-1$. This will be discussed in more detail in Section IV-A.

Our problem formulation holds several advantages over (5). First, note that (8) has one differential state fewer than (5). Therefore, it contains fewer optimization variables. Second, the optimal control problem has a fixed (pseudo) end-time, namely $s = 1$, which simplifies the problem to great extent. Third, because s is the independent variable, the optimization variables no longer enter the problem through $y_r(s)$. Instead, only the coefficients of the terms in Φ_b and Ψ_b depend on the reference path. Therefore, $y_r(s)$ will have a much smaller influence on the geometry of the feasible set compared to formulation (5).

Problem formulation (8) can be seen as a generalization of the path following problem for robotic manipulators, described in [2]. In their formulation, $k = 2$, Φ_b and Ψ_b are linear in the optimization variables, $\mathcal{X}$ and $\mathcal{U}$ are convex and only $F(\beta, w)$ that result in a convex objective are allowed. For this specific case, the optimization problem is convex and is solved globally and efficiently. As we will illustrate in Section V, the more general problem can also be solved efficiently but global optimality cannot be guaranteed in general.

A. Singularities

In order for the system to perform a rest-to-rest transition, the initial and terminal states must be set to zero. This could be accomplished by imposing zero initial and terminal values for β in (8). However, these constraints will likely result in the integral (7) being undefined, as shown in the proposition below.

Proposition 1. *If $b(0) = 0$, $\partial_s b(0) = 0$ and $\partial_s^2 b(0) = c < \infty$ then the integral*

$$\int_0^1 \frac{1}{\sqrt{b(s)}} ds$$

is undefined.

Proof: Consider $b(s)$ near 0. Since $\partial_s^2 b(0) = c$, there always exists an η such that $b(s) \leq cs^2$ on $[0, \eta]$ or, equivalently

$$\frac{1}{\sqrt{b(s)}} \geq \frac{1}{\sqrt{c}s} \text{ for } s \in [0, \eta].$$

Consequently

$$\int_0^\eta \frac{1}{\sqrt{b(s)}} ds \geq \int_0^\eta \frac{1}{\sqrt{c}s} ds = \infty,$$

when $c < \infty$. Hence the integral $\int_0^1 \frac{1}{\sqrt{b(s)}} ds$ is undefined. ■

In other words, from $k \geq 3$ onwards, zero initial or terminal values for β render (7) undefined and hence cannot be imposed. Therefore, rest-to-rest transitions are imposed indirectly by choosing a suitable parametrization for $y_r(s)$. Indeed, by ensuring $\partial_s^i y_r(0) = 0$ for $i = 1, \ldots, k-1$, we can impose $\partial_t^i y_r(s(0)) = 0$ regardless of the value of $\beta(0)$ (cf. (4)). Any path $y_r(s)$ can easily be reparameterized by $y_r(p(s))$ with $p(\cdot)$ an odd degree polynomial of sufficient degree such that $p(0) = 0$, $p(1) = 1$, $\partial_s p(s) \geq 0$ for $s \in [0, 1]$ and $\partial_s^i p(0) = \partial_s^i p(1) = 0$ for $i = 1, \ldots, k-1$. In the following section we use this reparameterization to prove under which conditions a given reference is exactly followable.

B. Path followability

Sufficient conditions to determine whether a given geometric reference is exactly followable are already derived in [7]. However, by using the proposed time transformation and a suitable reparameterization of the path these conditions allow for an intuitive understanding and a simplification of the proof. Therefore, we repeat them here.

Theorem 1 (Exact path followability)**.** *Assume that the system* (1) *starts and stops in steady state and the maps Φ_b and Ψ_b are continuous. If for all $s \in [0, 1]$*

$$\Phi_b(s, 0, \ldots, 0) \in \mathrm{int}(\mathcal{X}) \text{ and } \Psi_b(s, 0, \ldots, 0) \in \mathrm{int}(\mathcal{U}), \quad (9)$$

then $y_r(s)$ can be followed exactly by the flat system (1)*. Furthermore, the minimal transition time is finite for the time-optimal case.*

Proof: The proof relies on finding a feasible solution to (8). First, the path is reparameterized such that the system starts and stops in steady state, as described in the previous section.

Now choose $b(s) = \epsilon > 0$, where the constant ϵ is chosen small enough such that for all $s \in [0, 1]$, due to (9) and the continuity of the maps Φ_b and Ψ_b,

$$\Phi_b(s, \epsilon, 0, \ldots, 0) \in \mathcal{X}$$

and

$$\Psi_b(s, \epsilon, 0, \ldots, 0) \in \mathcal{U}.$$

Hence, $b(s) = \epsilon$ yields a feasible solution to (8). Moreover, when considering the time-optimal case, i.e. $F(\cdot) = 0$, the minimal transition time $T^* \leq 1/\sqrt{\epsilon}$ is finite. ■

Intuitively, the theorem states that $y_r(s)$ is exactly followable if each point of the reference path can be visited in steady state while staying in the interior of the constraint set. The proof shows this can be accomplished by traveling with very low constant speed $b(s) = \epsilon$ along the path.

V. Examples

The robotic manipulator and the quadrotor serve as illustration for our framework. We consider a robotic manipulator with and without viscous friction in the joints. In the latter case the optimization problem is convex. We show that calculation times remain comparable for both cases. For highly nonlinear systems as the quadrotor, solutions are also calculated efficiently without requiring an accurate user-defined initial guess. In all presented simulations, we rely on automatic initialization of the solver.

The problems are discretized using direct transcription and are modeled in CasADi [19]. Ipopt [20] is used as solver, using exact Hessians and auto initialization. The problems are solved on a 2.66 GHz PC with 4 GB of RAM memory. The computation times are reported as the sum of the time spent in Ipopt and in nonlinear functions calls. By using C code generation, the latter could be sped up to a factor ten. To simplify the definition of the problems, a software package, released under LGPL, is developed and can be downloaded from http://www.kuleuven.be/optec/software/. For future benchmarking, the code used to solve the presented examples is included.

A. Path following for a robotic manipulator

The equations of motion of an n-DOF robotic manipulator with joint angles $q \in \mathbb{R}^n$, can be written as a function of the applied joint torques $\tau \in \mathbb{R}^n$ [21]

$$\tau = D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q),$$

where $D(q)$ is a positive definite mass matrix, $C(q, \dot{q})$ is a matrix accounting for Coriolis and centrifugal effects and $g(q)$ denotes the gravity vector. Verscheure et al. [2] make the critical assumption that $C(q, \dot{q})$ is linear in the joint velocities, which renders their problem convex. However, when a matrix $B(q)$ containing viscous terms is added, we find

$$\tau = D(q)\ddot{q} + (C(q, \dot{q}) + B(q))\dot{q} + g(q).$$

Now $C(q, \dot{q}) + B(q)$ is affine in $\dot{q}$ and convexity of the path following problem is destroyed.

In this example, we consider a two-link planar elbow manipulator in a vertical plane, as in Fig. 1, with and without a diagonal viscous friction matrix $B(q)$, which allows us to compare the convergence of

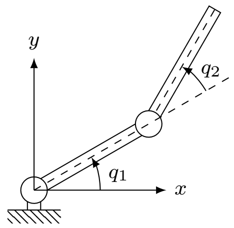

Fig. 1. Planar elbow manipulator with joint q_1 and q_2

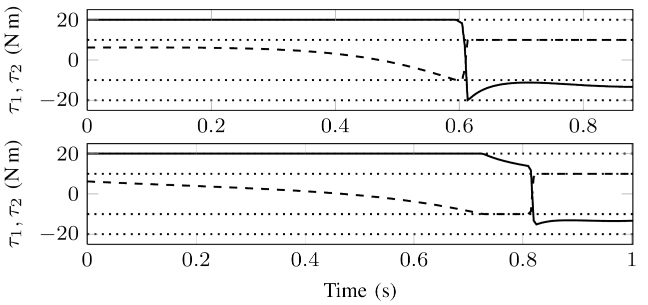

Fig. 2. Optimized torques (τ_1: solid, τ_2: dashed) for a planar elbow manipulator with (bottom) and without (top) viscous friction in the joints following the reference (10)

both the convex and nonconvex problem for the same problem size. Obviously, the joint angles $(q_1, q_2)^T$ are a flat output for this system. We consider the geometric reference for both joints

$$y_r(s) = \left(\frac{\pi}{2}s, -\pi s\right)^T, \tag{10}$$

such that for a robot with equal link lengths the end-effector traces a line along the x-axis. In order to start with zero joint velocity the reference is reparameterized as described in Section IV-A. The joint torques are constrained to

$$\tau \in [-20, 20]\ \mathrm{N\,m} \times [-10, 10]\ \mathrm{N\,m}.$$

Furthermore, a time-optimal, i.e. $F = 0$, is determined.

The discretized problem contains 400 variables, 199 equality and 400 inequality constraints. The solutions of both the convex and nonconvex problem are obtained in 24 iterations or 0.12 s. For the convex problem 0.036 s are spent in Ipopt and 0.084 s in nonlinear function calls, whereas for the nonconvex problem the computation times are 0.032 s and 0.088 s respectively. In our approach there is little difference in computation times whereas in [4], where a sequential convex programming approach is followed, the cost for solving the nonconvex problem is approximately four times the cost of the convex problem.

The optimal actuator torques are shown in Fig. 2. Both for the convex (top) and nonconvex (bottom) problem, at least one of the constraints is active at each time step, which is a typical property for time-optimal solutions. Also note that the execution time for the nonconvex problem is slightly larger due to the viscous friction in the joints.

The convergence of problem (8) is compared to (5) by increasing the nonlinearity of the reference path for the case without viscous friction. To this end, we define a nonlinear recursive function

$$r_i(s) = \frac{1}{1 + e^{\cos \pi r_{i-1}(s)}}, \tag{11}$$

with $r_0(s) = s$ and the reference path $y_r(s) = (r_i(s), -2r_i(s))$. Table I compares the number of iterations for different values of i for

i	Problem (5)	Problem (8)
1	97	28
2	38	32
3	78	30
4	79	32
5	91	31
6	94	32
7	64	34
8	–	32
9	105	32
10	189	34

TABLE I
NUMBER OF ITERATIONS FOR FORMULATIONS (5) AND (8) OF THE PATH FOLLOWING PROBLEM WITH PATH $y_r(s) = (r_i(s), -2r_i(s))$ AND $r_i(s)$ AS DEFINED IN (11)

both problem formulations. It is clear that our approach consistently outperforms (5). Moreover, the number of required iterations hardly changes, whereas for (5) it varies strongly with an increasing trend for increasing i, though with some outliers. For $i = 8$, the solver was unable to converge for formulation (5). These findings confirm that, contrary to [7], in our approach the nonlinearity of the path has little influence on the convergence.

B. Path following for a quadrotor

Aggressive, time-optimal maneuvers for quadrotors along a predetermined path can be calculated efficiently in our framework, as illustrated in this example. Fig. 3 shows the notation used. The coordinate $(x, y, z)^T$ represents the coordinate of the quadrotor's center of gravity with respect to a world frame and the XYZ Euler angles $(\phi, \theta, \psi)^T$ denote the roll, pitch and yaw angle. A flat output for the quadrotor is $(x, y, z, \psi)^T$. The reader is referred to [22], for further information concerning flatness. The control input of the quadrotor $u = (u_1, u_2, u_3, u_4)^T$ consists of the net body force u_4 and the three body moments u_1, u_2, u_3 and can also be related to the angular velocities of the rotors [22].

A geometric reference trajectory is usually planned only in $(x, y, z)^T$ and therefore allows some freedom in the reference trajectory for the yaw angle ψ. One possibility is to align one of the quadrotor's arm with the tangent dy/dx, i.e. $\psi = \arctan \frac{\partial_s y(s)}{\partial_s x(s)}$. In this example we consider following reference

$$\begin{aligned} y_r(s) = (&\cos(2\pi s), \sin(2\pi s), \\ &(0.9(e^s - 1) + 0.1\sin(2\pi s))^2, 2\pi s)^T \end{aligned} \tag{12}$$

The reference is reparameterized as described in Section IV-A such that the quadrotor's initial and terminal states are zero. The inputs are constrained to

$$u \in [-8, 8]\ \mathrm{N\,m} \times [-8, 8]\ \mathrm{N\,m} \times [-8, 8]\ \mathrm{N\,m} \times [1, 32]\ \mathrm{N}.$$

A time-optimal, i.e. $F = 0$, solution is obtained in 22 iterations or 1.348 s, of which 0.416 s in Ipopt and 0.932 s in nonlinear function calls for a problem with 750 variables, 600 inequality and 596 equality constraints. The optimized control inputs are shown in Fig. 4. Note that at each time instant one of the constraints is active. Fig. 5 illustrates the position and orientation of the quadrotor for 20 equidistant time steps.

Subsequently, a term is added to the objective function that penalizes the energy consumed by the thrust force. The objective function is formulated as

$$J = \frac{T}{T^*} + \frac{\gamma}{E^*} \int_0^T u_4(t) v(t)\ dt, \tag{13}$$

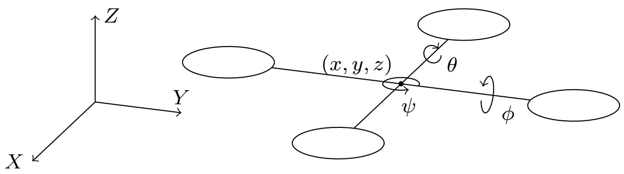

Fig. 3. The quadrotor with coordinates (x, y, z) of the center of gravity and roll, pitch, yaw angles (θ, ϕ, ψ)

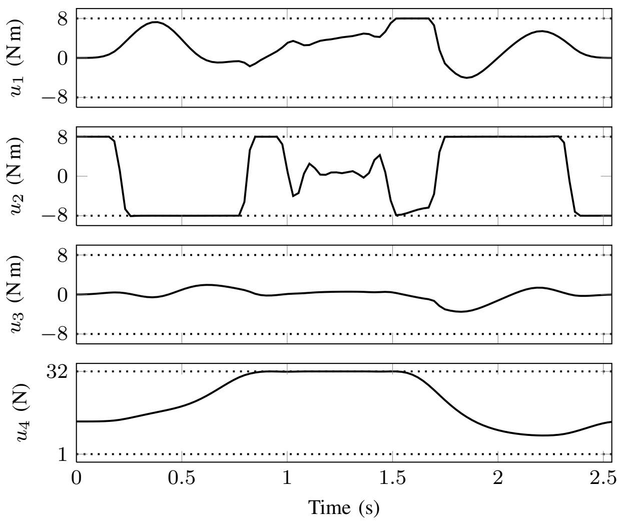

Fig. 4. Time-optimal control inputs for a quadrotor following (12)

where T^* denotes the minimal motion time and E^* is the corresponding energy consumed by the thrust force, γ determines the trade-off between time optimality and energy optimality and $v(t)$ is the absolute velocity of the quadrotor. Figure 6 illustrates the time-energy trade-off by varying γ from 0 to 10. As expected, an increase in execution time results in a decreased energy consumption. Figure 7 illustrates the control forces and torques for increasing γ as a function of the path coordinate s. The higher γ the smoother the control signals are, resulting in higher energy efficiency and less wear of the actuators. The calculation times remain comparable to the time-optimal problem: 23 iterations or 1.648 s of which 0.676 s spent in Ipopt and 0.972 s in nonlinear function calls. Note the slight increase in the solver time

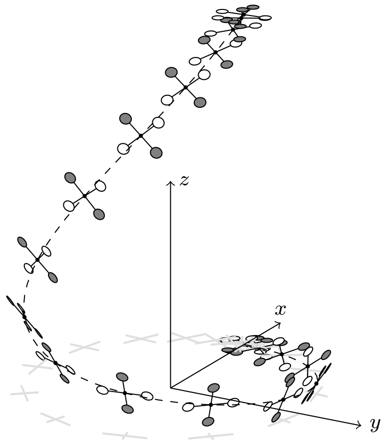

Fig. 5. Quadrotor's position and orientation in 20 equidistant time steps

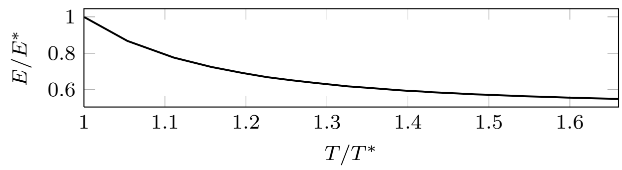

Fig. 6. The time-energy trade-off by varying γ in (13) from 0 to 10

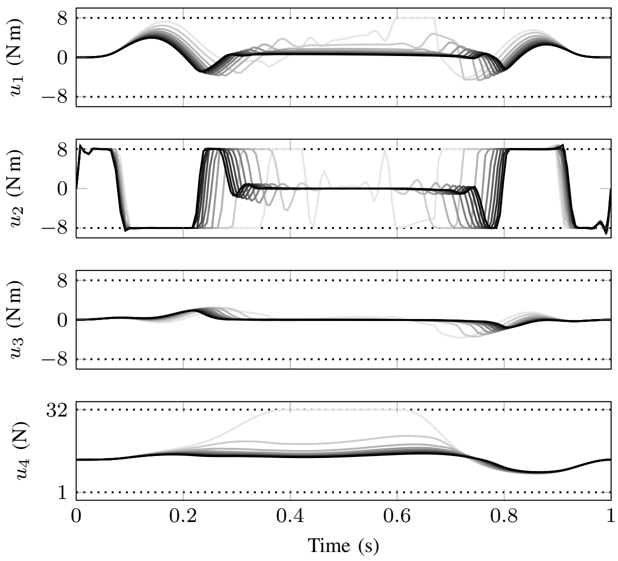

Fig. 7. Time-energy optimal control inputs for varying γ for a quadrotor following (12)

due to the increased nonlinearity of the objective function.

VI. Discussion

The applications in robotics surpass the two examples of a robotic manipulator and a quadrotor. Assuming a no-slip condition, most mobile vehicles are differentially flat, with a physical coordinate of the vehicle as flat output (see for example [11], [23]), making them excellent candidates for the proposed framework. Also the overhead crane, consisting of a variable length pendulum attached to a trolley that can move horizontally, is a classical example of differentially flat systems [11]. Its flat output is the coordinate of the load. Again, this system is ideally suited for path following of the flat output. These examples illustrate our method's relevance for robotic applications. Moreover, many systems from other research domains are also differentially flat, for example the continuous stirred tank and fed-batch reactors [7], [15], [24], [25] in chemical applications and an induction motor in electrical applications [15], [26].

Both numerical examples illustrate that solutions are obtained efficiently within a few CPU seconds, making the method practical in academic and industrial practice. Moreover, we rely on the automatic initialization of Ipopt and the user is not required to supply an initial guess, as opposed to [27] where for quadrotors a time-consuming genetic algorithm is used to generate an initial guess. The proposed problem formulation (8) also allows for other objectives than time, such as energy consumption, and arbitrary constraints, such as torque rate constraints in robotic applications, which cannot be included in the framework of [2].

Compared to the problem formulation (5) from [7], we believe our geometric problem formulation has to two important advantages. First, the problem is transformed into a fixed end-time problem, hereby simplifying the problem to a large extent. Second, the optimization variables no longer enter the problem through $y_r(s)$, as the path coordinate s becomes the independent variable instead of the time. Numerical experiments illustrate that our formulation shows little dependence on the reference path as opposed to [7].

Acknowledgment

The authors would like to thank Joris Gillis for his help with the software implementation and Gijs Hilhorst for useful discussion on the proof of Proposition 1.

References

[1] K. Shin and N. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Transactions on Automatic Control*, vol. 30, no. 6, pp. 531–541, 1985.

[2] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: a convex optimization approach," *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2318–2327, 2009.

[3] J. Bobrow, S. Dubowsky, and J. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, 1985.

[4] F. Debrouwere, W. Van Loock, G. Pipeleers, Q. Tran Dinh, M. Diehl, J. De Schutter, and J. Swevers, "Time-optimal path following for robots with convex–concave constraints using sequential convex programming," *Robotics, IEEE Transactions on*, vol. 29, no. 6, pp. 1485–1495, 2013.

[5] R. Skjetne, T. I. Fossen, and P. V. Kokotović, "Robust output maneuvering for a class of nonlinear systems," *Automatica*, vol. 40, no. 3, pp. 373–383, 2004.

[6] G. M. Hoffmann, S. L. Wasl, and C. J. Tomlin, "Quadrotor helicopter trajectory tracking control," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, (Honolulu, Hawaii), 2008.

[7] T. Faulwasser, V. Hagenmeyer, and R. Findeisen, "Optimal exact path-following for constrained differentially flat systems," in *Proceedings of 18th IFAC World Congress*, (Milano, Italy), 2011, pp. 9875–9880.

[8] Z. K. Nagy, "Model based robust control approach for batch crystallization product design," *Computers & Chemical Engineering*, vol. 33, no. 10, pp. 1685–1691, 2009, Selected Papers from the 18th European Symposium on Computer Aided Process Engineering (ESCAPE-18).

[9] C. Raczy and G. Jacob, "Fast and smooth controls for a trolley crane," *Journal of Decision Systems*, vol. 8, no. 3, pp. 367–388, 1999.

[10] J. Hauser and A. Saccon, "A barrier function method for the optimization of trajectory functionals with constraints," in *Proceedings of the IEEE Conference on Decision and Control*, (San Diego, CA, USA), 2006, pp. 864–869.

[11] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of nonlinear systems: introductory theory and examples," *International Journal of Control*, vol. 61, pp. 1327–1361, 1995.

[12] J. Levine, *Analysis and control of nonlinear systems: a flatness-based approach*. Springer, 2010.

[13] H. J. Sira Ramírez and S. K. Agrawal, *Differentially flat systems*. Marcel Dekker, 2004, 467 pp.

[14] N. Petit, M. B. Milam, and R. M. Murray, "Inversion based constrained trajectory optimization," in *Proceedings of 5th IFAC symposium on nonlinear control systems (NOLCOS)*, (St. Peterburg, Russia), 2001.

[15] P. Martin, R. M. Murray, and P. Rouchon, "Flat systems, equivalence and trajectory generation," Centre Automatique et Systèmes, École des Mines de Paris, 2003.

[16] M. B. Milam, K. Mushambi, and R. M. Murray, "A new computational approach to real-time trajectory generation for constrained mechanical systems," in *Proceedings of the IEEE Conference on Decision and Control*, (Sydney, Australia), vol. 1, 2000, pp. 845–851.

[17] D. Henrion and J. B. Lasserre, "LMIs for constrained polynomial interpolation with application in trajectory planning," *Systems & Control Letters*, vol. 55, no. 6, pp. 473–477, 2006.

[18] N. Faiz, S. K. Agrawal, and R. M. Murray, "Trajectory planning of differentially flat systems with dynamics and inequalities," *Journal of Guidance, Control and Dynamics*, vol. 24, no. 2, pp. 219–227, 2001.

[19] J. Andersson, J. Åkesson, and M. Diehl, "Casadi: a symbolic package for automatic differentiation and optimal control," English, in *Recent Advances in Algorithmic Differentiation*, vol. 87, 2012, pp. 297–307.

[20] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, 1 2006.

[21] M. Spong and S. Hutchinson, *Robot modeling and control*. Wiley, 2005, 496 pp.

[22] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (Shanghai, China), 2011, pp. 2520–2525.

[23] J.-C. Ryu and S. K. Agrawal, "Differential flatness-based robust control of mobile robots in the presence of slip," *The International Journal of Robotics Research*, vol. 30, no. 4, pp. 463–475, 2011.

[24] R. Rothfuss, J. Rudolph, and M. Zeitz, "Flatness based control of a nonlinear chemical reactor model," *Automatica*, vol. 32, no. 10, pp. 1433–1439, 1996.

[25] R. Mahadevan, S. K. Agrawal, and F. J. III, "Differential flatness based nonlinear predictive control of fed-batch bioreactors," *Control Engineering Practice*, vol. 9, no. 8, pp. 889–899, 2001.

[26] P. Martin and P. Rouchon, "Flatness and sampling control of induction motors," in *Proc. IFAC World Congress*, 1996, pp. 389–394.

[27] L. Lai, C. Yang, and C. Wu, "Time-optimal control of a hovering quad-rotor helicopter," *Journal of Intellegent and Robotic Systems*, vol. 45, no. 2, pp. 115–135, 2006.